AMENDMENTS TO THE SPECIFICATION

Please replace the paragraph beginning on page 1, line 4, with the following rewritten paragraph:

--This application claims priority to U.S. Provisional Patent Application No. 60/151,531 entitled "SYSTEM AND METHOD FOR PROVIDING COMPUTER SECURITY" filed August 30, 1999, which is incorporated herein by reference for all purposes~~, and to~~. U.S. Patent Application No. 09/615,697 entitled "SYSTEM AND METHOD FOR COMPUTER SECURITY" filed July 14, 2000, ~~which~~ is incorporated herein by reference for all purposes.--

Please replace the paragraph beginning on page 1, line 10, with the following rewritten paragraph:

-- This application is related to co-pending U.S. Patent Application No. 09/651,439 ~~No. _____ (Attorney Docket No. RECOP011)~~ entitled SYSTEM AND METHOD FOR DETECTING COMPUTER INTRUSIONS filed concurrently herewith, which is incorporated herein by reference for all purposes; and co-pending U.S. Patent Application No. 09/651,303 ~~No. _____ (Attorney Docket No. RECOP012)~~ entitled EXTENSIBLE INTRUSION DETECTION SYSTEM filed concurrently herewith, which is incorporated herein by reference for all purposes; and co-pending U.S. Patent Application No. 09/651,854 ~~No. _____ (Attorney Docket No. RECOP013)~~ entitled SYSTEM AND METHOD FOR USING LOGIN CORRELATIONS TO DETECT INTRUSIONS filed concurrently herewith, which is incorporated herein by reference for all purposes; and co-pending U.S. Patent Application No. 09/651,304 ~~No. _____ (Attorney Docket No. RECOP015)~~ entitled SYSTEM AND METHOD FOR ANALYZING FILESYSTEMS TO DETECT INTRUSIONS filed concurrently

herewith, which is incorporated herein by reference for all purposes; and co-pending U.S. Patent Application No. 09/651,306 ~~No. _____ (Attorney Docket No. RECOP016)~~ entitled SYSTEM AND METHOD FOR DETECTING BUFFER OVERFLOW ATTACKS filed concurrently herewith, which is incorporated herein by reference for all purposes; and co-pending U.S. Patent Application No. 09/654,347~~No. _____ (Attorney Docket No. RECOP017)~~ entitled SYSTEM AND METHOD FOR USING TIMESTAMPS TO DETECT ATTACKS filed concurrently herewith, which is incorporated herein by reference for all purposes.--

Please replace the paragraph beginning on page 74, line 1, with the following rewritten paragraph:

-- In an embodiment of the invention, the analysis engine approaches the problem by cross-checking the available sources of signatures, and issuing a multi-level assessment of whether that file is suspected of having been maliciously changed, as in the example illustrated in Figure 9.--

Please replace the paragraph beginning on page 74, line 4, with the following rewritten paragraph:

--One check is to iterate through the files in the package management database, comparing the signatures in the database to the signature of the current version of the file (902 and 904). If the signatures match, the analysis engine draws no conclusion, because this provides no evidence to distinguish the two cases: (1) the file could be correct; or (2) the attacker has modified the database to have the signature of a file he installed. If there is a mismatch of signatures, the analysis engine then checks if the mismatch is expected (906), and if so, the file is

evaluated as legitimate (910); if not, the file is flagged as suspicious (908). Expected mismatches are determined by a set of rules:

- Package management systems allow the package creator to place files in various categories. If the file is in one of the categories regarded as changeable (*e.g.*, configuration files, log files), ignore the mismatch. However, since the categorization is dependent on the efforts of the package creator and mis-categorizations are common, a file not being in one of these categories is not strong evidence of a problem.
- If the file size in the package management database is zero, assume that it is a logging file.
- Attempt to match the suffix on the file against commonly used suffixes for files expected to change. For example, "*.conf*", "*.config*", "*.log*".
- Compare the location of the file against conventions for where changeable files are place. For example, the directories */etc* and */var/lib* are common locations for configuration files and configurable scripts, and */var/log* is a traditional location for log files.
- Compare against an internal database of known exceptions.

Another check is to compare signatures for files listed in the internal database of signatures (912). This database is a combination from multiple sources:

- Some software vendors publish signatures for their products.
- Signatures computed from installed copies of the software
- by the manufacturer of the inventive system for inclusion in the intrusion detection system distribution

  by the customer for applications installed at his site--